

CLAIMS:

We claim:

1. An autonomic memory leak detection and remediation system comprising:
an autonomic garbage collector coupled to memory configured to store object instances which can be accessed by executing processes and which can be referenced by other object instances in said memory;
a tracing policy coupled to said autonomic garbage collector, said tracing policy specifying an aging threshold for a number of garbage collection passes during which an object instance in said memory is considered a loiterer when said object instance had not been accessed by one of said executing processes.
2. The system of claim 1, wherein said memory is a heap managed through a virtual machine.
3. The system of claim 1, wherein said autonomic garbage collector comprises a mark and sweep garbage collector modified both to manage aging values associated with object instances in said memory and also to compare said aging values to said aging threshold to identify loiterers.
4. The system of claim 1, wherein said tracing policy further comprises a specification for at least one action to be undertaken upon detecting a loiterer.

5. The system of claim 1, wherein said tracing policy further comprises a listing of exempt classes based upon which object instances are exempted from being labeled loiterers.

6. A method for detecting and remediating a memory leak, the method comprising the steps of:

establishing an aging value for an object instance created in memory;
resetting said aging value when said object instance is referenced by an executing process;
incrementing said aging value during a garbage collection pass when said object instance had not been referenced by an executing process since a previous garbage collection pass; and,
when said aging value exceeds a threshold value, processing said object instance as a loiterer.

7. The method of claim 6, wherein said establishing step further comprises the steps of:

locating equivalent object instances in said memory; and,
processing said equivalent object instances in said memory as loiterers.

8. The method of claim 6, wherein said processing step comprises at least one of clearing at least one cache in memory, and reporting said object instance as a loiterer in a log file.

9. The method of claim 6, further comprising the step of foregoing said processing step where said object instance belongs to a specified exempt class.

10. An autonomic memory leak detection and remediation method comprising the steps of:

modifying a mark and sweep garbage collection process to manage aging values associated with object instances created in memory; and,

processing as loiterers selected ones of said object instances having aging values which exceed a predetermined threshold.

11. The method of claim 10, wherein said processing step comprises the step of processing as loiterers selected ones of said object instances not belonging to an exempt class where said selected ones of said object instances have aging values which exceed a predetermined threshold.

12.. The method of claim 10, wherein said processing step comprises clearing at least one cache in memory, and reporting said object instance as a loiterer in a log file.

13. A machine readable storage having stored thereon a computer program for detecting and remediating a memory leak, the computer program comprising a routine set of instructions for causing the machine to perform the steps of:

establishing an aging value for an object instance created in memory;

resetting said aging value when said object instance is referenced by an executing process;

incrementing said aging value during a garbage collection pass when said object instance had not been referenced by an executing process since a previous garbage collection pass; and,

when said aging value exceeds a threshold value, processing said object instance as a loiterer.

14. The machine readable storage of claim 13, wherein said establishing step further comprises the steps of:

locating equivalent object instances in said memory; and,

processing said equivalent object instances in said memory as loiterers.

15. The machine readable storage of claim 13, wherein said processing step comprises clearing at least one cache in memory, and reporting said object instance as a loiterer in a log file.

16. The machine readable storage of claim 13, further comprising the step of foregoing said processing step where said object instance belongs to a specified exempt class.